LAVARNET: Neural Network Modeling of Causal Variable Relationships for Multivariate Time Series Forecasting

Christos Koutlis*, Symeon Papadopoulos, Manos Schinas, Ioannis Kompatsiaris

Information Technologies Institute, Centre of Research and Technology Hellas, Thessaloniki, Greece

Abstract

Multivariate time series forecasting is of great importance to many scientific disciplines and industrial sectors. The evolution of a multivariate time series depends on the dynamics of its variables and the connectivity network of causal interrelationships among them. Most of the existing time series models do not account for the causal effects among the system's variables and even if they do they rely just on determining the between-variables causality network. Knowing the structure of such a complex network and even more specifically knowing the exact lagged variables that contribute to the underlying process is crucial for the task of multivariate time series forecasting. The latter is a rather unexplored source of information to leverage. In this direction, here a novel neural network-based architecture is proposed, termed LAgged VAriable Representation NETwork (LAVARNET), which intrinsically estimates the importance of lagged variables and combines high dimensional latent representations of them to predict future values of time series. Our model is compared with other baseline and state of the art neural network architectures on one simulated data set and four real data sets from meteorology, music, solar activity, and finance areas. The proposed architecture outperforms the competitive architectures in

*Corresponding author

Email addresses: ckoutlis@iti.gr (Christos Koutlis), papadop@iti.gr (Symeon Papadopoulos), manosetro@iti.gr (Manos Schinas), ikom@iti.gr (Ioannis Kompatsiaris)

most of the experiments.

Keywords: multivariate time series forecasting, machine learning, neural networks, connectivity network

1. Introduction

Time series forecasting is a research topic that attracts great interest in many areas such as meteorology [1], finance [2, 3], seismology [4], energy consumption [5, 6], and traffic [7]. Adequately modeling the evolution patterns of time series and thus making accurate estimations of their future values can provide us with crucial information such as warnings about an upcoming storm, earthquake, stock price increase, or traffic jam. Besides, not only the time dependence between past and future values is vital for a system's evolution but also the causal interrelationships among its coupled variables, which might occur in a non-uniform manner [8, 9, 10]. Other studies have also pinpointed the importance of non-uniform embeddings of multivariate time series in forecasting [11].

One of the most well known and widely adopted time series forecasting models, ARIMA [12], captures linear correlations between past and future values. Also, many other machine learning methodologies have been employed to serve the same purpose, such as k nearest neighbors [13], Gaussian processes [14], random forests [15], multi-layer perceptrons [16] and deep belief networks [17]. However today state of the art results in sequence modeling have been produced by recurrent neural network (RNN) architectures [18], which are known to capture the non-linear time dependence between the predicted future value and the preceding values. The standard form of a recurrent neural network is proposed by Elman in [19] as in Equation 1:

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h) \tag{1a}$$

$$y_t = \sigma_y(W_y h_t + b_y) \tag{1b}$$

where x_t is the input time series at time t, h_t is the hidden state of the network

at time t, y_t is the network's output at time t and W_h, U_h, b_h, W_y, b_y are trainable variables. Also, σ_h and σ_y are non-linear activation functions.

More complex and effective architectures, both in natural language processing and in time series forecasting[1], have been proposed since then. Cho et al. [20] proposed the encoder-decoder architecture in which an RNN is used to encode the input sequence and a second RNN is then used to decode the encoder's output and make the final prediction. In [21] the latter idea is enhanced by an attention mechanism between the encoder and the decoder, which forces the decoder to focus on the most relevant time steps of the input sequence. Some alternative attention mechanisms have also been proposed during the past few years [22, 23, 24]. Other network architectures based on RNNs and Long Short-Term Memory networks (LSTM) [25] have been deployed considering both time directions on the input data [26, 27], skip connections between layers [28], autoregressive components [29, 30], multi-level attention mechanisms [31] and missing values handling [32, 23]. Also, convolutional neural network (CNN) architectures [33, 34] and architectures that combine CNNs with other models [35, 36] have been proposed for regression and forecasting of time series.

However, while all of these architectures capitalize on the estimation of time dependence and global information extraction, none of them accounts for the causal interdependence among the coupled variables of the underlying multivariate mechanism. Recent studies though paved the way towards this direction by employing dual-stage attention mechanisms [37, 38], which apply attention weights on the input variables during the encoding phase and then apply attention weights on the time steps during the decoding phase. We consider the dual-stage attention-based recurrent neural network (DARNN) [37] as a state of the art model in our comparative study presented in the results section.

To the best of our knowledge, although the literature numbers loads of forecasting methodologies none of them takes into account the importance of certain

[1] These two disciplines are subdivisions of the more general term *sequence modeling* and many related methodologies can be applied to both.

lagged variables in the system's evolution mechanism. Notwithstanding, it is reasonable to consider that extracting more fine-grained information can lead to better forecasting accuracy. For instance, if one variable affects the target after τ time steps and another variable after $\tau + h$ time steps, knowing only that these two variables affect the target (or even not knowing it) is probably less helpful than being aware of the lags as well.

In our approach, hidden states are generated by the model as high dimensional latent representations of the multivariate time series' lagged variables, for each pair of variable and time step and not just for each time step. Then, trainable weights are applied to the representations which ideally will tend to foster the correct lagged variables and enable the model to mine this very wanted knowledge of coupling structure. To this end, we changed Elman's equations by introducing also the variable information in the model in addition to the time step information. Three model versions are proposed here, one that does not consider previous hidden states and consequently being non-recurrent, termed LAgged VAriable Representation NETwork (LAVARNET), one that considers the previous hidden state of the corresponding variable, termed Recurrent LAgged VAriable Representation NETwork (R-LAVARNET) and one that considers the previous hidden states of all variables, termed Fully Recurrent LAgged VAriable Representation NETwork (FR-LAVARNET).

We conducted a series of experiments using one simulated data set from the well-known difference equations system coupled Hénon maps [39, 40] and four real data sets from meteorology, music, solar activity, and finance areas. The results show that the proposed architecture is capable of making multivariate and univariate forecasts based on multivariate input signals with great accuracy. Additionally, it outperforms other baseline and state of the art neural network architectures in most of the experiments. Also, a second simulation study reveals the interpretable nature of our model, in which it is shown that the lagged variables that contribute to the system's evolution are fostered by the corresponding trainable weights. Finally, the authors consider as the main contributions of this paper the following:

- A novel neural network-based architecture is proposed for multivariate time series modeling and forecasting. It considers multivariate input and either multivariate or univariate output depending on the under study problem

- The main advantage of this method is that estimates the underlying coupling structure among the measured variables of the system and exploits the information gained by the most important lagged variables. Thus, its behavior is interpretable providing extra information regarding the underlying mechanism as the weights of lagged variables estimated at training phase reveal the causal relationships among the measured variables

- Until now most forecasting methods do not account for the causality patterns among the measured variables and if they do they estimate patterns at variable granularity level. Our method estimates even more fine-grained causality patterns at lagged variable granularity level for the first time

- This architecture is found to exhibit superior forecasting behavior, compared to other baseline and state of the art models, on one simulated data set and on three out of four real data sets considered in this study

The rest of the paper is structured as follows. In Section 2 the problem formulation and the proposed architecture are presented, in Section 3 the data sets are described, in Section 4 the experiments are elaborated and in Section 5 conclusions are given.

2. Methodology

2.1. Problem formulation

The problem of multivariate time series forecasting is formulated as follows. Consider a series of measured signals, $\mathbf{X}=[x_{1,:}, x_{2,:} \ldots, x_{T,:}]$ with $x_{i,:} \in \mathbb{R}^K$ $i = 1, 2, \ldots, T$, where K is the number of variables and T is the number of time steps. The goal is to predict $x_{T+1,:}$ if all variables are of interest or $x_{T+1,k}$ if only variable k is of interest, given $\mathbf{X}$.

In our approach, the estimation of the importance of lagged variables in predicting the future is vital so we would like to clarify the meaning of this notion. Considering a process $x(t)$ at time $t \in \mathbb{N}$, its lagged variables are defined as $x(t-\tau)$, where $\tau \in \mathbb{N}$ is the, so called, lag. In the multivariate case for instance, if $x_{T+1,k}$ is caused by $x_{T,k}$ and $x_{T-1,k-1}$, then variable k at lag τ=1 (namely $x_k(t-1) \equiv x_{T,k}$) and variable $k-1$ at lag τ=2 (namely $x_{k-1}(t-2) \equiv x_{T-1,k-1}$) are the responsible lagged variables for the evolution of variable k.

2.2. LAVARNET: Lagged Variable Representation Network

Here, a time series forecasting model is proposed that is based on Elman's equations for the recurrent neural network (Equation 1). The drawback of recurrent neural network architectures that we alleviate here is that they do not account for interrelationships among the time series' variables and hence lack knowledge regarding the underlying causality network of the system. The causal relationships among variables of a coupled multivariate system determine its evolution (along with other factors such as self-dependencies of each variable), making this information crucial for the task of forecasting. To address this problem we add to Elman's equations a term that holds the variable information in addition to the term that holds the time step information when the hidden states are generated. This procedure produces a larger number of hidden states $T \cdot K$ (where T is the number of time steps and K is the number of variables) than all classic recurrent neural networks, that produce T hidden states. On one hand, this requires more memory but on the other hand, additional useful information is leveraged.

As we have already mentioned three model versions are proposed, one that does not consider previous hidden states and consequently can be considered as non-recurrent, termed LAgged VAriable Representation NETwork (LAVARNET), one that considers the previous hidden state of the corresponding variable, termed Recurrent LAgged VAriable Representation NETwork (R-LAVARNET) and one that considers the previous hidden states of all variables, termed Fully

Recurrent LAgged VAriable Representation NETwork (FR-LAVARNET).

For the definition of the proposed model, we begin by determining the equations for the hidden states' generation. The LAVARNET's equations are:

$$h_{t,k} = \sigma_h(W_T \cdot x_{t,:} + W_V \cdot x_{:,k} + b_h) \tag{2a}$$

$$y_{t,k} = \sigma_y(W_y \cdot h_{t,k} + b_y) \tag{2b}$$

The R-LAVARNET's equations are:

$$h_{t,k} = \sigma_h(W_T \cdot x_{t,:} + W_V \cdot x_{:,k} + U_h \cdot h_{t-1,k} + b_h) \tag{3a}$$

$$y_{t,k} = \sigma_y(W_y \cdot h_{t,k} + b_y) \tag{3b}$$

Finally, the FR-LAVARNET's equations are:

$$h_{t,k} = \sigma_h(W_T \cdot x_{t,:} + W_V \cdot x_{:,k} + \tilde{U}_h \cdot h_{t-1,:} + b_h) \tag{4a}$$

$$y_{t,k} = \sigma_y(W_y \cdot h_{t,k} + b_y) \tag{4b}$$

where $x_{t,:} \in \mathbb{R}^K$ is the multi-variate input at time t, $x_{:,k} \in \mathbb{R}^T$ is the multi-time-step input of variable k, $h_{t,k} \in \mathbb{R}^n$ is the hidden state for variable k at time t with n being the number of neurons, $y_{t,k} \in \mathbb{R}^n$ is the output vector for variable k at time t, $h_{t-1,:} \in \mathbb{R}^{n \cdot K}$ is the concatenation of all hidden states (all variables) of the previous time step and W_T $(n \times K)$, W_V $(n \times T)$, U_h $(n \times n)$, $\tilde{U}_h$ $(n \times n \cdot K)$, b_h $(n \times 1)$, W_y $(n \times n)$, b_y $(n \times 1)$ are trainable variables. Also, σ_h and σ_y are non-linear activation functions. More specifically here the sigmoid activation is used for both σ_h and σ_y. Then, a matrix of trainable weights $A = \{a_{t,k}\}$ is defined, with $t = 1, \ldots, T$ and $k = 1, \ldots, K$ and each scalar $a_{t,k}$ is multiplied with the corresponding output vector $y_{t,k}$. The weights corresponding to important lagged variables should take non-zero values (either positive or negative) and the weights corresponding to non-important lagged variables should take zero (or as close as possible to zero) values after the training procedure. Finally, all output vectors are concatenated in one vector and passed through dense layers for the prediction of future values of the time series. In the case of predicting just one variable's future values, there is only one fully connected layer in the

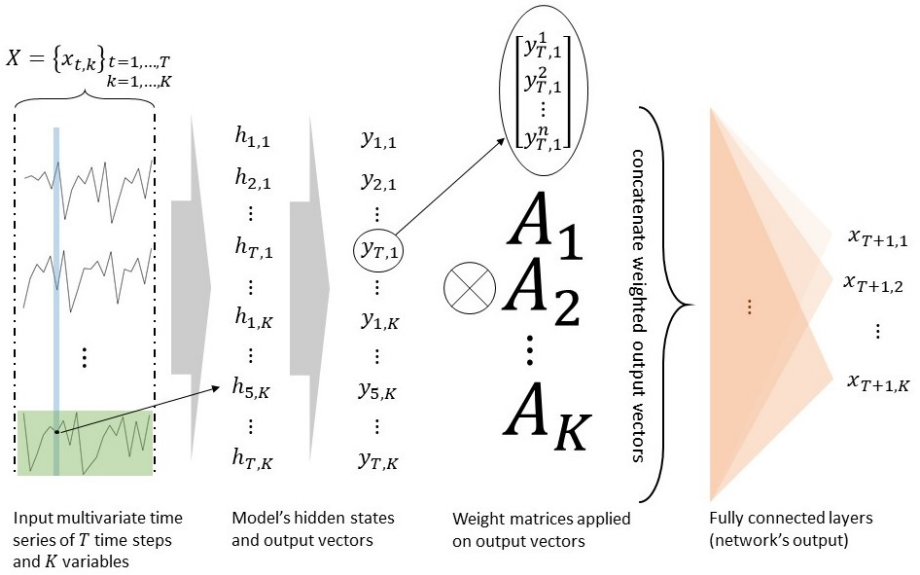


Figure 1: Architecture of LAVARNET. Each lagged variable $x_{t,k}$ is represented by a hidden state $h_{t,k}$ which is then transformed to the output vector $y_{t,k}$. Consequently, trainable weights are applied on all output vectors and finally dense layers are employed for the forecasts.

network's output and only one trainable matrix of weights A, while in the case of predicting many or all of the system's variables multiple independent fully connected layers are employed as output layers. Additionally, in the latter case, multiple trainable matrices of weights $A_1, \dots, A_K$ (with $A_i = \{a^i_{t,k}\}$ and $i = 1, \dots, K$) are considered, one for each target i. The previous step is of utmost importance as each target might be driven by different lagged variables which should be fostered accordingly.

In Figure 1, a graphical representation of our model is illustrated for better comprehension. Each lagged variable $x_{t,k}$ is transformed into the hidden state $h_{t,k}$ through the Equations 2a (LAVARNET), 3a (R-LAVARNET) or 4a (FR-LAVARNET) and then the hidden state $h_{t,k}$ is transformed into the output vector $y_{t,k}$ through the Equations 2b (LAVARNET), 3b (R-LAVARNET) or 4b (FR-LAVARNET). For the prediction of the first variable ($k = 1$) each of the $T \cdot K$ output vectors $y_{t,k}$ is multiplied by the corresponding element $a^1_{t,k}$ of the

trainable matrix A_1 and consequently, all these new vectors are concatenated in one vector of $T \cdot K \cdot n$ elements, where n is the user-defined number of neurons. The prediction for $x_{T+1,1}$ is then calculated by a fully connected layer. The predictions $x_{T+1,k}$ for the rest variables $k = 2, \ldots, K$ are performed accordingly using a different matrix A_k and different fully connected layer (network's output) for each variable k.

As one may notice, although before the fully connected layers the model exploits all time steps up to time step T, the network's output concerns time step $T + 1$, hence there is no information leakage from future to past. The main reason that the proposed architecture performs that well in forecasting is that it is able to capture the connectivity structure of the underlying complex mechanism that generates the measurements. Moreover, it is able to estimate accurately not only the subsystems that contribute to the evolution of each system variable but also the exact lagged variables. This fact makes the results (and consequently the proposed model's behavior) interpretable which is a major advantage and in Section 4.5 a simulation study is presented to showcase this. Finally, we selected as starting point the simple RNNs equations, because any other ideally preferable choice such as Gated Recurrent Unit (GRU) [20] or LSTM [25] would dramatically increase the number of estimated parameters and thus make the training of the architecture infeasible.

3. Data

3.1. Simulated data

The simulated data for the forecasting task are generated by the well-known Hénon map system [39] and more precisely the coupled Hénon maps system as defined in [40] with the chain connectivity pattern among its variables. Many simulation scenarios are considered involving different number of variables K=5,10,15, number of time steps T=3,5,10,15 and time series lengths L=200, 500, 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000 and 10000.

For each simulation scenario (e.g. K=5, T=10 and L=1000) 5 Monte-Carlo simulations are generated in order to obtain average model performance.

For the interpretability simulation study presented in Section 4.5 we use the linear stochastic process of vector autoregressive model (VAR) [41] with a random network of connections [42] among its variables for the generation of multivariate time series. The time series length is set to L=5000 and different simulation scenarios in terms of number of variables K=2,3,...15 and model order P=1,2,3 are considered. Also, 10 Monte-Carlo simulations per simulation scenario are generated for the estimation of average model performance.

3.2. Real data

For the real data analysis four data sets from different research domains are considered, one data set related to weather (SML2010)[2], a set of data related to musical genre popularity (GenrePopularity) that is generated as part of the FuturePulse project[3], a data set related to solar activity (Solar-Energy)[4] and a data set related to finance (Currency)[5].

The SML2010 data set contains indoor temperature time series and other relevant quantities like Carbon dioxide in ppm and sunlight in the south facade. This data set is collected from a monitor system mounted on a domestic house. Our target variable is the room temperature and 16 other relevant driving series are used as input to our models as well. The data were sampled every minute and was smoothed with 15-minute means. In this study, we use the first 3200 data points as the training set, the following 400 data points as the validation set, and the last 537 data points as the test set.

The GenrePopularity data set contains time series data from 2000-01-01 until 2019-10-31 related to the popularity level of 60 musical genres (presented in Table 1) in four countries: Great Britain, United States of America, Sweden

[2]https://archive.ics.uci.edu/ml/datasets/SML2010

[3]http://www.futurepulse.eu/

[4]https://www.nrel.gov/grid/solar-power-data.html

[5]https://www.kaggle.com/thebasss/currency-exchange-rates

African	Alternative	Ambient	Americana	Bass
Blues	Breakbeat	Children's Music	Christian	Christmas
Classical	Country	Dance/EDM	Reggaeton	Death Metal
Disco	Doom Metal	Downbeat	Drum&Bass	Dubstep
Electronic	Electronica	Experimental	Folk	Funk
Garage	Hardcore	Hard Rock	Heavy Metal	Hiphop
House	Techno	Indie	Industrial	Inspirational
Instrumental	Jazz	Karaoke	Latin	Lounge
Mariachi	Metal	Musical	Opera	Pop
R&B	Reggae	Rock	Rockabilly	Salsa
Samba	Singer-Songwriter	Soul	Soundtrack	Spoken Word
Surf	Tango	Tech House	Thrash Metal	TripHop

Table 1: Musical genres considered in the GenrePopularity data set.

and Canada. Each time series point concerns the percentage of entries, in charts of a certain country, related to a specific musical genre for a sliding time window of 4 weeks with a step of one week. We have collected data from 60 charts in Great Britain, 116 charts in the United States of America, 26 charts in Sweden, and 18 charts in Canada. Then by aggregating the entries, 4 multivariate time series are generated with 60 variables and 1031 time points each. Also, a moving average filter of order 4 is applied for noise reduction. Training, validation, and test sets are generated by splitting the time series into 618, 206, and 207 time points respectively. Most of the time series variables are sparse, thus for the forecasting task all genre time series with more than 100 zeros are discarded from the models' input. As target variables, we selected three non-sparse musical genres namely Pop, Rock, and Hip-hop.

The Solar-Energy data set contains time series data about the solar power production records in the year of 2006, which is sampled every 10 minutes from 137 photovoltaic plants in Alabama State. The total number of time points is 52,560 and we split it into training, validation, and test sets with 31,536, 10,512 and 10,512 time points respectively. As target variables, we use the first 10 variables, after sorting the file names.

The Currency data set contains the daily currency exchange rates as reported to the International Monetary Fund by the issuing central bank. Included are 51 currencies over the period from 01-01-1995 to 11-04-2018. The format is known as currency units per U.S. Dollar. Explained by example, each rate in the Euro column says how much U.S. Dollar you had to pay at a certain date to buy 1 Euro. Hence, the rates in the column U.S. Dollar are always 1. We use data from 30-10-1998 which is the date Euro was released, currencies with more than 1500 missing values and currencies with constant exchange rate are discarded and the rest are linearly interpolated. So finally, the data set contains 41 currencies and 4,986 time points which are split into 2991 training samples, 997 validation samples, and 998 test samples. As target variables we use the first ten[6] currencies being Australian Dollar (1), Botswana Pula (2), Brazilian Real (3), Brunei Dollar (4), Canadian Dollar (5), Chilean Peso (6), Chinese Yuan (7), Colombian Peso (8), Czech Koruna (9), Danish Krone (10).

As a pre-processing step z-score normalization is applied to SML2010, Solar-Energy and Currency, while no normalization is applied to GenrePopularity which contains values between 0 and 1 by default.

4. Experiments

4.1. Competitive models

In order to perform a comparative study we consider three baseline and two state of the art time series models as competitive models. The three baselines are (1) the k nearest neighbors regression model (KNN), (2) the single layered recurrent neural network (RNN) [19] and (3) the single layered long short-term memory network (LSTM) [25] each followed by a dense output layer. The two state of the art models are (1) the dual-stage attention-based recurrent neural network (DARNN) [37], which considers an encoder with attention on the input

[6] We skip the second currency Bahrain Dinar as it exhibits constant exchange rate in the test set.

variables and a decoder with attention on the time steps, and (2) the WaveNet [43] as proposed in [44] for time series forecasting, which is mainly a stack of dilated convolutional layers with residual and skip connections.

For KNN, 5 neighbors are used and for RNN and LSTM, 128 neurons are considered in all real data experiments as baseline selections. For DARNN, the optimal number of neurons is determined after grid search among 32, 64, 128 neurons and for LAVARNET grid search among 5, 10, 20[7] is also employed. Finally, for WaveNet 64 filters and filter width 2 are considered for each of 6 stacked dilated convolutional layers with dilation rates 1, 2, 4, 1, 2, 4 respectively.

4.2. Training

For the training of all neural network-based models we used Adam optimizer and mean squared error loss function provided by TensorFlow 1.8.0. Also, three GPU devices (two GeForce GTX 1080 and one GeForce GTX 1070) are employed for the experiments. Additionally, for the KNN training the scikit-learn Python package implementation is employed.

In the experiments, the data sets were split into training (60%), validation (20%) and test (20%) sets (except for the SML2010 data set in which we opted for the same splitting as in [37] for comparison purposes). So, the models' training is performed on the training set, all models are checkpointed based on their performance on the unseen data of the validation set and their performance is evaluated on the test set.

Also, for DARNN we opt for the proposed in [37] learning rate strategy, starting with 0.001 and reducing by 10% every 10,000 iterations. For RNN, LSTM, and WaveNet we opt for a constant learning rate equal to 0.001 and for LAVARNET cosine annealing [45] is used, where at epoch i the learning rate

[7] Except for the Currency data set in which 32, 64 and 128 neurons are selected for the grid search because higher values produced better results in this data set.

$\eta(i)$ is set to:

$$\eta(i) = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})\left(1 + cos(\frac{i \cdot \pi}{E})\right) \quad (5)$$

where η_{max}=0.01, η_{min}=0.0001 and E is the number of epochs.

4.3. Evaluation

For the evaluation of all models on the forecasting task we use the error function mean absolute error (MAE) as in Equation 6:

$$MAE_k = \frac{1}{N}\sum_{t=1}^{N} \mid x_{t,k} - \hat{x}_{t,k} \mid \quad (6)$$

where k is the target variable, N is the number of samples in the test set, $x_{t,k}$ is the actual measurement and $\hat{x}_{t,k}$ is the prediction. In the case of multivariate prediction the average MAE_k across all k is considered.

For the interpretation of the results in terms of correct weighting of lagged variables (Section 4.5), we use the percentage of true driving lagged variables that were among the ones with the highest absolute weights assigned by LAVARNET. More precisely, in the simulations the exact lagged variables that drive each target variable is known. Let us say L_k is the set of lagged variables that drive target variable k. Then, for that target variable we determine the set of lagged variables $\tilde{L_k}$ that are associated with the $C(L_k)$ (where $C(S)$ is the cardinality of set S) highest absolute values of matrix A_k. Finally, using Equation 7, we compute the success percentage of true driving lagged variables of the whole system that were categorized by LAVARNET as such and use this score as evaluation index:

$$R_L = \frac{\sum_k C(L_k \cap \tilde{L_k})}{\sum_k C(L_k)} \quad (7)$$

where k is the target variable.

Also, we consider another less strict score namely the percentage of true driving variables that were categorized by LAVARNET as such. It is denoted by R_V and computed as in Equation 8:

$$R_V = \frac{\sum_k C(V_k \cap \tilde{V_k})}{\sum_k C(V_k)} \quad (8)$$

model	MAE
LAVARNET	**0.0430**
R-LAVARNET	0.0442
FR- LAVARNET	0.0460
LSTM	0.0534
RNN	0.0561
KNN	0.1473

Table 2: Average performance of models on the simulation data set, across all scenarios and Monte-Carlo simulations.

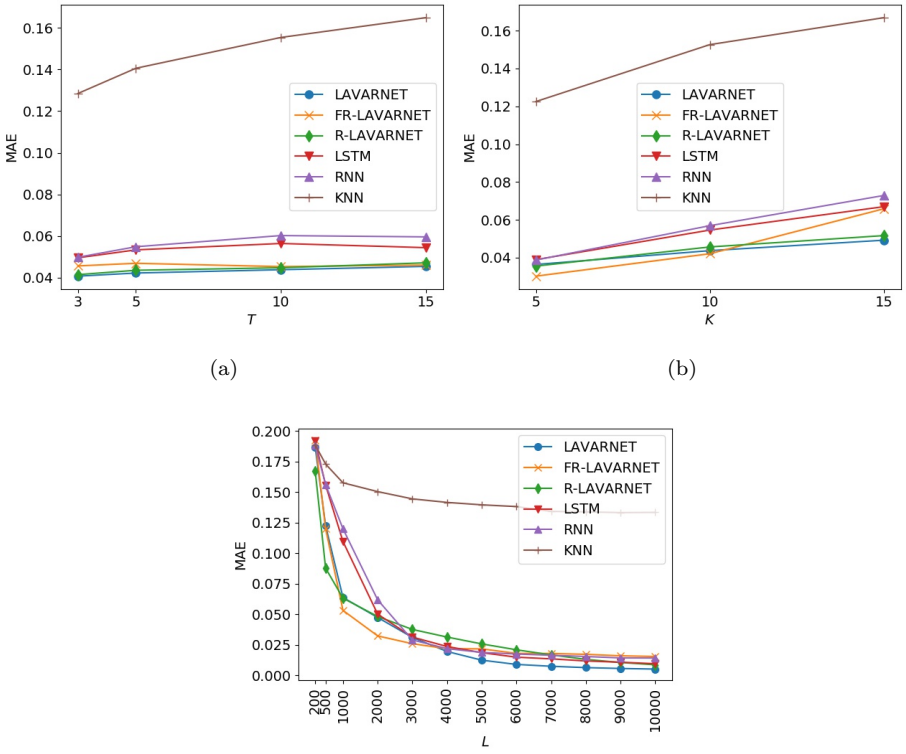
where k is the target variable, V_k is the set of variables that drive target variable k and $\tilde{V_k}$ is the set of variables that are associated with the $C(L_k)$ highest absolute values of matrix A_k.

4.4. Results

First, we present results for the simulation study in which our model is compared with the baseline models KNN, single layered RNN and single layered LSTM. In the simulation study we evaluate forecasting models in multivariate prediction, thus DARNN and WaveNet are discarded as not applicable[8]. Also, 100 neurons are considered for all neural network-based models' hidden state vectors in the simulation study. As described in Section 3.1 many simulation scenarios in terms of number of variables, number of time steps and time series length are considered as well as multiple Monte-Carlo simulations of each scenario. In Table 2, the average models' performance across all different scenarios and Monte-Carlo simulations is presented and the model exhibiting the best performance is highlighted with bold letters, being LAVARNET.

For more details Figure 2 is also provided. In this figure the average performance of the models is illustrated excluding the parameter (a) time steps, (b)

[8]DARNN model is designed to make univariate forecasts given multivariate signals as input and WaveNet is disigned to model univariate signals.



(a) (b)

(c)

Figure 2: The performance of LAVARNET, R-LAVARNET, FR-LAVARNET, LSTM, RNN and KNN in terms of mean absolute error (MAE) on multivariate prediction task for the simulated coupled Hénon maps system. (a) Average performance across number of variables, time-series length and Monte-Carlo simulations, (b) average performance across number of time steps, time-series length and Monte-Carlo simulations and (c) average performance across number of time steps, number of variables and Monte-Carlo simulations.

model	MAE
LAVARNET	0.0674±0.012
R-LAVARNET	**0.0389±0.012**
FR-LAVARNET	0.0804±0.013
LSTM	0.0849±0.007
RNN	0.0967±0.010
DARNN	0.0533±0.004
WaveNet	0.0866±0.0.0253
KNN	0.4983

Table 3: Evaluation of models on SML2010 data set using mean absolute error (MAE). The average value of the error function is presented along with the corresponding standard deviation. The model exhibiting the best performance is highlighted with bold letters.

number of variables and (c) time series length, respectively. It is observed that all models' performance decreases as the number of time steps and number of variables increases and that all models' performance increases as the time series length increases, as expected. The proposed models perform better than the baselines in all scenarios but LAVARNET and R-LAVARNET exhibit a more consistent behavior, especially when the number of variables is high. This is actually expected given that FR-LAVARNET's number of parameters is rapidly increasing with the number of variables. Additionally, R-LAVARNET produces the smallest errors when the time series length is small, while FR-LAVARNET and LAVARNET perform best when the time series length is of intermediate and large size respectively.

In Table 3, the results for the comparative study on the real data set SML2010, are presented. Apparently, R-LAVARNET produces the smallest errors and although overlapping, the intersection of uncertainty intervals with the second one, being DARNN, is very small. The WaveNet's performance is comparable to the performances of RNN, LSTM and FR-LAVARNET in this experiment and KNN produces the greatest errors.

In Table 4, the performance of forecasting models on the task of predicting musical genres' popularity is presented. We consider 12 different settings each of them related to a different country and a different target variable (musical genre). Also, the presented results are the average and standard deviation of MAE across 10 repetitions of each training/testing procedure. It is observed that in 7 settings LAVARNET performs best, in 2 settings R-LAVARNET performs best, in one setting the simple RNN model performs best and only in 2 settings DARNN outperforms all the other models. WaveNet and KNN yield the least accurate forecasts in most of the settings of this data set, with KNN producing the largest errors.

For the Solar-Energy data set that contains time ordered measurements from 137 variables, 10 comparative experiments are conducted forecasting future values of each of the first 10 variables, after sorting the file names. A multivariate prediction experiment would leave DARNN out of the comparative study and also would be infeasible in terms of GPU memory consumption. Additionally, conducting 137 separate experiments is computationally very costly, thus we opted for the first 10 variables. In Table 5 the corresponding results are illustrated, in which R-LAVARNET performs best in all 10 experiments. LAVARNET and LSTM produce comparable to R-LAVARNET's errors and all the rest models exhibit worse performance in the Solar-Energy data set.

		GB	US	SE	CA
Pop	LAVARNET	**0.0049±0.0003**	0.0100±0.0021	0.0087±0.0016	0.0110±0.0019
	R-LAVARNET	0.0060±0.0003	0.0160±0.0022	**0.0075±0.0005**	0.0100±0.0023
	FR-LAVARNET	0.0087±0.0007	0.0280±0.0049	0.0083±0.0008	0.0114±0.0022
	LSTM*	0.0114±0.0038	0.0180±0.007	0.0108±0.004	0.0096±0.004
	RNN*	0.0116±0.0033	0.0312±0.010	0.0124±0.006	**0.0089±0.003**
	DARNN	0.0092±0.0005	**0.0077±0.0005**	0.0086±0.0001	0.0114±0.0008
	WaveNet	0.0174±0.0127	0.0509±0.0237	0.0228±0.0053	0.0175±0.0076
	KNN	0.0649	0.0761	0.0536	0.0175
Rock	LAVARNET	0.0068±0.0004	**0.0029±0.0001**	**0.0017±0.0003**	0.0030±0.0005
	R-LAVARNET	0.0076±0.0006	0.0034±0.0004	0.0022±0.0003	**0.0028±0.0005**
	FR-LAVARNET	0.0103±0.0007	0.0044±0.0007	0.0026±0.0005	0.0048±0.0020
	LSTM	0.0114±0.003	0.0150±0.005	0.0050±0.001	0.0050±0.002
	RNN	0.0106±0.002	0.0137±0.007	0.0074±0.003	0.0055±0.002
	DARNN	**0.0062±0.0004**	0.0043±0.0002	0.0026±0.00009	0.0041±0.0003
	WaveNet	0.0209±0.0115	0.0096±0.0057	0.0197±0.0331	0.0088±0.0043
	KNN	0.0276	0.0133	0.0384	0.0116
Hip-hop	LAVARNET	**0.0031±0.0007**	**0.0039±0.0003**	**0.0023±0.0004**	**0.0048±0.0014**
	R-LAVARNET	0.0040±0.0004	0.0042±0.0005	0.0029±0.0004	0.0065±0.0011
	FR-LAVARNET	0.0054±0.0011	0.0065±0.0012	0.0049±0.0004	0.0102±0.0014
	LSTM	0.0108±0.0022	0.0131±0.0042	0.0067±0.0031	0.0112±0.0037
	RNN	0.0151±0.0044	0.0145±0.007	0.0063±0.0028	0.0132±0.004
	DARNN	0.0059±0.0006	0.0064±0.0020	0.0053±0.00009	0.0088±0.0013
	WaveNet	0.0102±0.0047	0.0159±0.0080	0.0089±0.0027	0.0134±0.0078
	KNN	0.0142	0.0134	0.0146	0.0331

Table 4: Evaluation of models on musical genre popularity forecasting task for three different musical genres (Pop, Rock, Hip-hop) as target in four different countries (Great Britain; GB, United States; US, Sweden; SE, Canada; CA). The evaluation index is MAE (the standard deviation is also presented) and the model exhibiting best performance at each setting is highlighted with bold letters.

k	LAVARNET	R-LAVARNET	FR-LAVARNET	LSTM	RNN	DARNN	WaveNet	KNN
1	1.18±0.09	**1.13±0.04**	1.33±0.13	1.40±0.27	1.43±0.24	1.53±0.08	1.57±0.23	2.26
2	1.30±0.10	**1.26±0.08**	1.50±0.18	1.38±0.12	1.51±0.18	1.65±0.04	2.06±0.38	2.71
3	0.38±0.03	**0.33±0.04**	0.42±0.05	0.42±0.10	0.44±0.09	0.49±0.02	0.67±0.34	0.92
4	0.39±0.03	**0.36±0.06**	0.44±0.06	0.47±0.09	0.50±0.09	0.54±0.05	0.64±0.21	0.94
5	0.39±0.03	**0.37±0.04**	0.43±0.06	0.38±0.04	0.45±0.07	0.53±0.03	0.67±0.16	0.97
6	0.87±0.09	**0.83±0.04**	0.92±0.11	0.97±0.16	1.00±0.10	1.05±0.04	1.14±0.14	1.60
7	0.39±0.04	**0.35±0.04**	0.41±0.03	0.43±0.09	0.49±0.10	0.49±0.04	0.56±0.10	0.92
8	0.36±0.03	**0.33±0.04**	0.40±0.07	0.37±0.07	0.39±0.06	0.47±0.03	0.45±0.05	0.93
9	0.38±0.02	**0.34±0.03**	0.43±0.05	0.37±0.04	0.43±0.08	0.52±0.04	0.63±0.21	0.98
10	0.37±0.03	**0.34±0.04**	0.44±0.09	0.39±0.04	0.47±0.09	0.51±0.03	0.63±0.21	0.92

Table 5: Evaluation of models on Solar-Energy data set for the first 10 variables. The evaluation index is MAE (the standard deviation is also presented) and the model exhibiting best performance at each experiment (target variable k) is highlighted with bold letters.

In Table 6, the results for the Currency data set are presented. For the same reasons as in the previous real data analysis we present results for the first 10 variables. In the results it is observed that our architectures exhibit the best performance only in 5 out of 10 experiments, while DARNN outperforms the other models in the rest 5 experiments. The baseline models KNN, RNN and LSTM do not outperform the other models in any experiment as expected. Also, the WaveNet outperforms LAVARNET in 3 out of 10 experiments, but in none of them exhibits the best performance across all models.

k	LAVARNET	R-LAVARNET	FR-LAVARNET	LSTM	RNN	DARNN	WaveNet	KNN
1	**0.011±0.005**	0.018±0.005	0.018±0.008	0.047±0.024	0.040±0.013	0.024±0.008	0.014±0.002	0.029
2	**0.008±0.003**	0.012±0.006	0.014±0.004	0.047±0.028	0.138±0.096	0.009±0.004	0.031±0.020	0.051
3	**0.049±0.007**	0.060±0.020	0.224±0.077	0.287±0.146	0.367±0.183	0.102±0.036	0.146±0.048	0.376
4	0.069±0.007	0.092±0.012	0.049±0.020	0.126±0.121	0.069±0.034	**0.026±0.015**	0.033±0.015	0.072
5	0.022±0.004	0.024±0.006	0.029±0.011	0.112±0.063	0.106±0.066	**0.017±0.003**	0.061±0.048	0.065
6	13.74±5.07	**9.75±7.72**	11.38±3.38	66.27±32.27	25.05±17.53	10.18±4.14	18.44±24.46	43.35
7	0.330±0.017	0.361±0.020	0.338±0.021	0.348±0.165	0.353±0.178	**0.069±0.026**	0.23±0.068	0.280
8	59.56±8.62	62.25±10.26	170.4±21.6	194.9±118.2	201.1±92.33	**47.43±17.86**	206.6±225.3	180.7
9	1.35±0.392	1.63±0.703	1.20±0.285	9.24±6.12	3.52±2.81	**0.751±0.097**	0.823±0.463	0.982
10	0.095±0.022	0.076±0.025	**0.074±0.026**	0.822±0.517	0.499±0.303	0.104±0.063	0.130±0.050	0.394

Table 6: Evaluation of models on Currency data set for the first 10 variables. The evaluation index is MAE (the standard deviation is also presented) and the model exhibiting best performance at each experiment (target variable k) is highlighted with bold letters.

τ \ k	1	2	3	4	5	6
1	**-0.172**	0.007	**-0.039**	-0.005	-0.007	**-0.142**
2	**-0.083**	**0.102**	-0.007	0.001	**0.137**	**-0.010**
3	**0.014**	**0.078**	**-0.012**	0.006	**-0.125**	**-0.095**

Table 7: Weights of lagged variables for the prediction of the first target variable of VAR(P=3) model with 6 variables assigned by LAVARNET (T=3) after training. With bold we denote the 12 highest weights in absolute value, k denotes the variable index and τ the lag.

Finally, the fact that FR-LAVARNET does not frequently perform better than LAVARNET and R-LAVARNET has a twofold explanation. The first reason is that FR-LAVARNET involves a much higher number of trainable parameters especially in cases of many coupled variables. Specifically, the matrix $\tilde{U}_h$ of Equation 4 has size $n \times n \cdot K$, while R-LAVARNET's corresponding matrix U_h has size $n \times n$ and LAVARNET does not even involve such a matrix. Second, FR-LAVARNET is likely to incorporate excessive or irrelevant information through the hidden states of the other variables at time step $t-1$.

4.5. Interpretability simulation study

Here, we use the VAR model for the generation of multivariate time series of different dimensions and model orders. The causal relationships among the variables of the system are selected at random using the Erdös-Rényi random network scheme with 40% network density and for each driving variable, all lags up to the model order are considered.

In Table 7 an example case of weights assigned by LAVARNET to the lagged variables[9] for brevity. of a multivariate time series is presented. More precisely, LAVARNET is trained on forecasting future values of a multivariate time series (generated by the VAR(P=3) model and having $K = 6$ variables) based on past values of all system variables and we present the weights of lagged variables that

[9] Actually, the weights are assigned to the model's output vectors $y_{t,k}$ that correspond to certain lagged variables as stated in the model's description, but this is omitted here

τ \ k	1	2	3	4	5	6
1	1	1	0	0	1	1
2	1	1	0	0	1	1
3	1	1	0	0	1	1

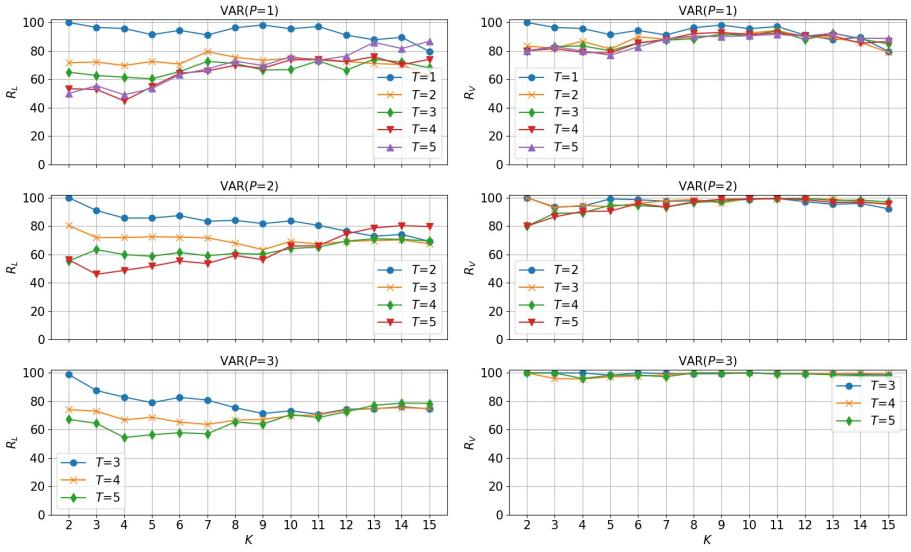
Table 8: Lagged variables of VAR(P=3) model with 6 variables, where the lagged variables that drive the first target variable are indicated by 1, the rest lagged variables are indicated by 0, k denotes the variable index and τ the lag.

correspond to the first target variable. In Table 8 the true lagged variables that contribute to the evolution of the first target variable of the system, are shown. As one can see, 10 out of 12 important lagged variables are assigned as such by LAVARNET and also it produces 2 mismatches, giving R_L=83.33% success percentage. Also, all 4 driving variables are correctly assigned as such, giving R_V=100%.

Aggregated results for all simulation scenarios and Monte-Carlo simulations are presented in Figure 3. In the simulation scenarios, it is observed that LAVARNET's average success percentage in identifying the correct lagged variables is close to 70%. While in the task of correctly identifying the driving variables it reaches values greater than 90% and even 100% in some cases. As expected, it is harder for LAVARNET to choose the correct lagged variables as the number of time steps T increases. Additionally, the latter is easier as the VAR order P increases, because the same variables contribute more intensely (with more lags). Interestingly, the number of variables K does not seem to adversely affect the correct identification of important lagged variables as it increases. On the contrary, less variability in success percentages with respect to different time steps T is exhibited as K increases.

4.6. Computational cost

Except for the forecasting accuracy, another aspect of a model's performance is computational cost. In Table 9, the average time required for model initialization and training on SML2010 data set, is illustrated for all neural network-based



(a) (b)

Figure 3: The average success percentage for (a) lagged variables (R_L) and (b) variables (R_V), across 10 Monte-Carlo simulations using LAVARNET for different VAR order P, number of time steps T and number of variables K.

model	time
LAVARNET	31.120 sec
R-LAVARNET	40.290 sec
FR-LAVARNET	33.796 sec
LSTM	30.562 sec
RNN	17.085 sec
DARNN	41.166 sec
WaveNet	101.865 sec

Table 9: The average time in seconds that it takes to form the graph and conduct the training on SML2010 data set per model.

models of our study and for the same number of epochs (70). Ten realizations are considered in order to present average performance. Also, one GPU device (GeForce GTX 1080) is employed for the computations.

The differences in execution time are not substantial among the competitive models, except RNN which is considerably faster and WaveNet which is considerably slower. However, RNN being fast comes as compensation for its poor forecasting accuracy. Among the models that perform well in forecasting LAVARNET is the fastest while FR-LAVARNET and R-LAVARNET follow right after leaving DARNN be the slowest.

5. Conclusions

In this work, we propose a novel neural network architecture that leverages intrinsically estimated high dimensional latent representations of lagged variables, to make multivariate time series forecasts. This model is evaluated on one simulated data set and four real data sets from meteorology, music, solar activity, and finance areas and it is found to outperform other baseline and state of the art neural network architectures and machine learning models in most of the experiments. Moreover, its behavior is interpretable by the trainable weights' values it assigns to lagged variables as it is shown by a separate simulation study.

However, our architecture did not exhibit superior performance across half of the experiments on the data set from finance, on which it performed up to the mark though. The state of the art neural network DARNN exhibits great performance in the rest experiments conducted on this data set and also WaveNet produces accurate forecasts but not the best among this ensemble of models. In the cases that DARNN outperforms LAVARNET, information from the multivariate signals is better exploited by DARNN in terms of connectivity estimation, in terms of temporal modeling or both. A plausible explanation to this might be that LAVARNET considers a stable over time causality network of the underlying mechanism that generates the measurements and in finance slight relative variability (or noise) might occur. On one hand, this might seem like a limitation, on the other hand knowing it is useful twofold (a) the user considers using it on suitable data sets or/and (b) an expert splits the data set into relatively stable (in terms of who is driving who) periods and the model is applied separately. In conclusion, this should not be a problem to the vast majority of data sets as all coupled systems preserve their coupling structure, either for long or for short periods, and at every phase transition the model can be re-trained.

Finally, the conducted experiments indicate that recurrent neural networks (even the baselines) are more powerful in temporal modeling and especially time series forecasting than convolutional based architectures (WaveNet) which still produce accurate predictions though. A combination of convolutional layers and LAVARNET seems like a promising extension of the current model that the authors will consider as future work. Future work will also focus on improving the proposed architecture in the direction of reducing memory consumption and computational cost.

Acknowledgments

This work is partially funded by the European Commission under the contract number H2020-761634 FuturePulse.

References

[1] G. Papacharalampous, H. Tyralis, D. Koutsoyiannis, Predictability of monthly temperature and precipitation using automatic time series forecasting methods, Acta Geophysica 66 (4) (2018) 807–831.

[2] O. B. Sezer, M. U. Gudelek, A. M. Ozbayoglu, Financial time series forecasting with deep learning: A systematic literature review: 2005-2019, Applied Soft Computing 90 (2020) 106181.

[3] B. M. Henrique, V. A. Sobreiro, H. Kimura, Literature review: Machine learning techniques applied to financial market prediction, Expert Systems with Applications 124 (2019) 226–251.

[4] G. Di Bello, V. Lapenna, M. Macchiato, C. Satriano, C. Serio, V. Tramutoli, Parametric time series analysis of geoelectrical signals: An application to earthquake forecasting in southern italy, Annali di Geofisica 39 (1) (1996) 11–21.

[5] L. Guo, L. Wang, H. Chen, Electrical load forecasting based on LSTM neural networks, in: International Conference on Big Data, Electronics and Communication Engineering (BDECE 2019), 2019, pp. 107–111.

[6] R. K. Jana, I. Ghosh, M. K. Sanyal, A granular deep learning approach for predicting energy consumption, Applied Soft Computing 89 (2020) 106091.

[7] B. L. Smith, B. M. Williams, R. K. Oswald, Comparison of parametric and nonparametric models for traffic flow forecasting, Transportation Research Part C: Emerging Technologies 10 (4) (2002) 303–321.

[8] L. Faes, G. Nollo, A. Porta, Non-uniform multivariate embedding to assess the information transfer in cardiovascular and cardiorespiratory variability series, Computers in Biology and Medicine 42 (3) (2012) 290–297.

[9] D. Kugiumtzis, Direct-coupling information measure from nonuniform embedding, Physical Review E 87 (2013) 062918.

[10] C. Koutlis, V. K. Kimiskidis, D. Kugiumtzis, Identification of hidden sources by estimating instantaneous causality in high-dimensional biomedical time series, International Journal of Neural Systems 29 (4) (2019) 1850051.

[11] S. Okuno, K. Aihara, Y. Hirata, Forecasting high-dimensional dynamics exploiting suboptimal embeddings, Nature: Scientific Reports 10 (664).

[12] G. E. P. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, CA, United States, 1970.

[13] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, The American Statistician 46 (3) (1992) 175–185.

[14] J. Yan, K. Li, E. Bai, Z. Yang, A. Foley, Time series wind power forecasting based on variant gaussian process and TLBO, Neurocomputing 189 (2016) 135–144.

[15] H. Tyralis, G. Papacharalampous, Variable selection in time series forecasting using random forests, Algorithms 10 (2017) 114.

[16] C. Hamzacebi, D. Akay, F. Kutay, Comparison of direct and iterative artificial neural network forecast approaches in multi-periodic time series forecasting, Expert Systems with Applications 36 (2) (2009) 3839 – 3844.

[17] T. Kuremoto, S. Kimura, K. Kobayashi, M. Obayashi, Time series forecasting using a deep belief network with restricted boltzmann machines, Neurocomputing 137 (2014) 47 – 56.

[18] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: Current status and future directions, arXiv:1909.00590.

[19] J. L. Elman, Finding structure in time, Cognitive Science 14 (2) (1990) 179–211.

[20] K. Cho, B. V. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), 2014, p. 1724–1734.

[21] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[22] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, V. Strijov, Position-based content attention for time series forecasting with sequence-to-sequence RNNs, in: International Conference on Neural Information Processing (ICONIP 2017), 2017, pp. 533–544.

[23] Y. G. Cinar, H. Mirisaee, P. Goswami, E. Gaussier, A. Aït-Bachir, Period-aware content attention RNNs for time series forecasting with missing values, Neurocomputing 312 (2018) 177–186.

[24] S. Shih, F. Sun, H. Lee, Temporal pattern attention for multivariate time series forecasting, Machine Learning 108 (8-9) (2019) 1421–1441.

[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.

[26] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing 45 (11) (1997) 2673–2681.

[27] S. Du, T. Li, Y. Yang, S. J. Horng, Multivariate time series forecasting via attention-based encoder-decoder framework, Neurocomputing.

[28] A. Graves, Generating sequences with recurrent neural networks, arXiv:1308.0850.

[29] Y. Y. Chang, F. Y. Sun, Y. H. Wu, S. D. Lin, A memory-network based solution for multivariate time-series forecasting, arXiv:1809.02105.

[30] G. Lai, W. C. Chang, Y. Yang, H. Liu, Modeling long and short-term temporal patterns with deep neural networks, in: SIGIR '18: The 41st International ACM SIGIR Conference on Research & Development in Information, 2018, pp. 95–104.

[31] Y. Liang, S. Ke, J. Zhang, X. Yi, Y. Zheng, GeoMAN: Multi-level attention networks for geo-sensory time series prediction, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), 2018, pp. 3428–3434.

[32] X. Tang, H. Yao, Y. Sun, C. Aggarwal, P. Mitra, S. Wang, Joint modeling of local and global temporal dynamics for multivariate time series forecasting with missing values, arXiv:1911.10273.

[33] A. Borovykh, S. Bohte, C. W. Oosterlee, Dilated convolutional neural networks for time series forecasting, Journal of Computational Finance, Forthcoming.

[34] I. Koprinska, D. Wu, Z. Wang, Convolutional neural networks for energy time series forecasting, in: 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8.

[35] H. J. Sadaei, P. C. de Lima e Silva, F. G. Guimaraes, M. H. Lee, Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series, Energy 175 (2019) 365 – 377.

[36] R. G. Cirstea, D. V. Micu, G. M. Muresan, C. Guo, B. Yang, Correlated time series forecasting using multi-task deep neural networks, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Association for Computing Machinery, 2018, pp. 1527–1530.

[37] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, G. Cottrell, A dual-stage attention-based recurrent neural network for time series prediction, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17), 2017, pp. 2627–2633.

[38] Y. Liu, C. Gong, L. Yang, Y. Chen, DSTP-RNN: A dual-stage two-phase attention-based recurrent neural network for long-term and multivariate time series prediction, Expert Systems with Applications 143.

[39] M. Hénon, A two-dimensional mapping with a strange attractor, Communications in Mathematical Physics 50 (1) (1976) 69–77.

[40] E. Siggiridou, C. Koutlis, A. Tsimpiris, D. Kugiumtzis, Evaluation of granger causality measures for constructing networks from multivariate time series, Entropy 21 (11) (2019) 1080.

[41] S. Basu, G. Michailidis, Regularized estimation in sparse high-dimensional time series models, The Annals of Statistics 43 (4) (2015) 1535–1567.

[42] P. Erdös, A. Rényi, On random graphs, Publicationes Mathematicae 6 (1959) 290–297.

[43] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, ArXiv abs/1609.03499.

[44] A. I. Borovykh, S. M. Bohte, C. W. Oosterlee, Conditional time series forecasting with convolutional neural networks, in: Lecture Notes in Artificial Intelligence, 2017, p. 729–730.

[45] I. Loshchilov, F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, in: 5th International Conference on Learning Representations (ICLR 2017), 2017.